Featherstone Primary School: Progression and Sequencing within Computing

Intent: Computing at Featherstone allows children to safely and responsibly work with a range of technology-based software and hardware to see the impact that this spectrum of technology has on our lives and the wider society. Children take part in learning with safe, real-life application processes so that they understand the artificial and digital systems that ease our lives, expressing their digital literacy through the use of ICT. Ultimately, children of Featherstone leave KS2 with knowledge of how to be safe participants of the digital world.

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Content Knowledge: Hardware and Software	Hardware: BeeBots Software: iPad App (Beebot)	Hardware: BeeBots Software: Turtle Logo and Scratch	Hardware: Sphero Software: Scratch	Software: Scratch and Python	Software: HTML web site and stop motion videos	Hardware: Crumble Software: Movie making
Content Knowledge: **Computer Science *(Knowledge of computers and how they work)*:** • **Algorithms** • **Programming**	Create simple programmes	Understand what algorithms are, how they are implemented and that programmes work by following precise and clear instructions Create simple programmes	Begin to write programmes to accomplish given goals	Consolidate and embed how to write programmes to accomplish given goals Use sequence, selection and repetitions in programmes and work with variables including various forms of input/output	Design and write programmes that accomplish given goals, incl. controlling or simulating physical systems Use and combine given software, on a range of devices, to accomplish given goals (collect, analyse, design, create, present and analyse)	Design and write programmes that accomplish intended goals independently chosen, including controlling or simulating physical systems Select, use and combine given software, on a range of devices, to accomplish given goals
Content Knowledge: **Computer Science *(Knowledge of computers and how they work)*:** • **Computational thinking *(Solving problems in Computing)***	Use logical reasoning to predict the behaviour of simple programmes	Debug simple programmes	Use logical reasoning to explain how some simple algorithms work With support, detect and correct errors in algorithms and programmes	Use logical reasoning to explain how some simple algorithms work With growing independence, detect and correct errors in algorithms and programmes	Begin to solve complex problems by breaking them down into smaller parts, with support or advice Independently detect and correct errors in algorithms and programmes	Solve more complex problems by breaking them down into smaller parts Take pre-emptive steps to begin to avoid errors in algorithms and programmes

Featherstone Primary School: Progression and Sequencing within Computing

Skills	**Information Technology** ***(The use of computers in different sectors and contexts)*** • **Digital Artefacts** ***(Digital objects incl. text, media, image and sound, created by humans)***	Use Y1 I.T. skills to create… • Using a trackpad or mouse • Switching on and shutting down a computer • Launch an application and manipulate (maximise, close and minimise) windows • Using an iPad to take a photo/record	Use Y2 I.T. skills to create… • Embedding Year 1 fundamentals • Drawing shapes • Shape fill • Shape outline • Font (style, colour, size) • Underlining and italics • Saving and retrieving a file • Creating a document	Use Y3 I.T. skills to create… • Embedding Year 2 fundamentals • Word Art • Spell check • **B** / <u>U</u> • Copy and paste information/pictures into a document • Re-sizing pictures • Beginning to use Microsoft PowerPoint and adding slides Use search technologies effectively and understand how results are shown (selected and ranked)	Use Y4 I.T. skills to create… • Embedding Y3 skills • Copy and paste using keyboard shortcuts (Ctrl + C and Ctrl + V) • Formatting picture • Creating a folder • "Dragging and dropping" a file into a folder • Using Microsoft PowerPoint to add animations/transitions	Use Y5 I.T. skills to create… • Embedding Y4 skills • Alt + Shift to alternate between windows • Drawing a table and inserting columns/rows • Bullet-points and text alignment • Formatting pictures • Applying all skills in a Microsoft document	Use Y6 I.T. skills to create… • Embedding Y5 skills • Manipulate windows for split screen (when combining software) • Watermark • Page insert/orientation • Embed table knowledge including merging cells • Applying all skills learnt in a Microsoft PowerPoint.
	Information Technology ***(The use of computers in different sectors and contexts)*** • **Computing Contexts** ***(How Computing is used purposefully)***	Recognise common uses of ICT around school	Recognise common uses of ICT beyond school	Understand the Google Classroom platform, incl. how it works (i.e. drive, stream, assignments).;	Understand computer networks incl. the internet, and how they can provide multiple services and opportunities	An overview of the history of computers and their impact on humanity, including: • Charles Babbage's 1833 Analytical Engine • 1st Generation of Computers 1940 – 1955 • 2nd Generation of Computers 1957 – 1963 • 3rd Generation of Computers 1964 – 1971 • 4th Generation of Computers 1971 – 1980 • 5th Generation of Computers 1980 onwards	An understanding of emerging technologies, including artificial intelligence and medical advancements, that are set to shape our future

Featherstone Primary School: Progression and Sequencing within Computing

	Concept	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Digital Literacy: Online safety *(Skills and knowledge required to be safe and effective)*	**Online behaviour**	Understand and explain the main consequences of unsafe/unwise online behaviour (i.e. someone might change their password or delete their work)		Understand the need and reasons to abide by the school's acceptable use policy/Google Chromebook Agreement Recognise acceptable and unacceptable behaviour, including "trolling"	Understand that the internet has its bad points as well as good points An introduction to social media apps and their potential dangers An introduction to online grooming, self-esteem and online attention	Embed social media apps and their potential dangers Online grooming (link to Stolen Lives)	Explain a person's online responsibility Evaluate all ICT decisions for e-safety before, during and after using the software/hardware In depth look at their digital footprint (specifically pictures online) Evaluate what social media apps children should be using and how to behave on them Online grooming, self-esteem and online attention (link to Stolen Lives)
	Applying online safety	Use technology respectfully and safely		Use technology respectfully, safely and responsibly Sign Year 3 e-safety pledge.	Use technology respectfully, safely and responsibly, including how to deal with "cyberbullying". Review e-safety pledge and sign Year 4 e-safety pledge.	Use technology respectfully, safely and responsibly, showing an awareness of their digital footprint Review previous year's safety pledge and sign Year 5 e-safety pledge.	Always use technology respectfully, safely and responsibly, showing an increased awareness of their digital footprint and how it can last Take ownership of, add to and sign Year 6 e-safety pledge.
	Personal information	Keep personal information private and understand that information on a computer can be shared elsewhere		Understand the difference between what information can and cannot be shared online	Know how to stay safe online at home as well as school		
	Support and reporting	Identify where to go for help or support when they have online concerns		Identity a range of ways to report computing concerns, including on online games and apps	Understand the benefits and drawbacks of social media, including how to identify and report concerns on social media	No additional online safety concepts. Recap if required to ensure learning is not lost	
	Evaluating online content	Evaluating online content not introduced at Key Stage One		An introduction to fake news	Understand that not all information on the internet is accurate (link to URLs) Evaluate digital content for utility/usefulness	Explain why some information on the internet is false (i.e. fake news) Evaluate digital content for bias	Evaluate all digital content for accuracy

Featherstone Primary School: Progression and Sequencing within Computing

Year 1 Computing Curriculum Guide

Content Knowledge	Explanation	Vocabulary	Potential journey/outcome
Use some logical reasoning to predict the behaviour of simple programmes	Logical reasoning means thinking logically or systematically to solve computational problems/scenarios. The best way to understand what a programme does, or to solve errors in a programme, is to think through sensibly what is supposed to happen. In Year 1, a programme would be a set of simple instructions. Telling something (a piece of technology) an instruction to do something. This would be a physical movement of a person to achieve something, at first. They would then use this understanding and apply it to a machine.	Forward Backwards Turn Left/Right Instructions What would happen if…? If I do this...	Children would spend a series of lessons exploring what a programme or series of instructions are. Most of this would involve the idea of giving instructions to a human/machine so that something happens. This may start with the whole class giving one child, or the teacher, a goal/mission to achieve. Maybe the teacher has to make her way through the class to reach the door. Children could potentially get the children to write down arrows on whiteboards. The children would think carefully (logically) before hitting the imaginary "GO!" button on the person doing the action. When I click forward three times and then turn, the Beebot will jump (this is illogical and not sensible). "Children, before we hit go, lets really think about what would happen if we said 3 steps forward. If I go three steps, would I be blocked?" If there is a mistake, children can go back and correct the mistake (this is called debugging but children don't need this until Year 2). Praise when the children think the steps through in their head- they are predicting. Praise good use of positional language. Children would then move on to planning a set of programmes/instructions to give to a robot. Children need to understand that the instruction that they give to their teacher to achieve a certain goal will be different if they were getting a machine to do the same thing. They would think logically/carefully/sensibly about what instructions to give and would test this as they go in a trial and error way. Children will have succeeded in this objective when they have predicted how a set of instructions behaves and ultimately reaches a goal. Task (Potentially 2-3 sessions): • Beebots travelling around a maze to get to a goal. • Beebots to visit different characters on a map in relation to an English text being studied.

Fundamental Use of Technology	Technology in the Wider Curriculum **Note: these are **not** computing objectives
Using a trackpad or mouse Switching on and shutting down a computer Launch an application and manipulate (maximise, close and minimise) windows Using an iPad to take a photo/record	<u>iPad camera</u> Children could use an iPad to record themselves making a smoothie. They could record the instructions like a cookery show and the teacher could play it back to the whole class when evaluating. <u>Green screen</u> Could the children use the green screen to perform a weather forecast in your outdoor topic? Green screen and editing would have to be done by the teacher. <u>Puppet Pals</u> Could the children use the Puppet Pals app to present information about Queen Elizabeth? 

Featherstone Primary School: Progression and Sequencing within Computing

Year 2 Computing Curriculum Guide

Content Knowledge	Subject Knowledge	Vocabulary	Potential journey/outcome
Understand what algorithms are, how they are implemented and that programmes work by following precise and clear instructions (with a recap on logical reasoning from Year 1).	Generally speaking, an algorithm is a set of precise instructions that a machine is given (input) to achieve an objective/goal (output). Children do not need to know the difference between an algorithm/programme so both these words can be used in reference to instructions. It **can** be explained by saying that an algorithm is the human understanding of what needs to be done. Programming is an algorithm that is put into a language that the machine can understand. E.g. The Beebot does not understand forward, forward, turn (algorithm) but the Beebot does understand Up arrow, Up arrow, Right arrow (programme).	Programme (the instructions given in a machine language) Algorithm (instructions given in a human language) Input Ouput Instructions Clear (Specific/Precise) Turtle Logo codes: • Fd 4(Forward 4) • Bk 7(Back 7) • Pd(pen down) • Rt (right turn) • Lt (left turn Both turns have to be followed by 90,180,270 depending on how far you want to turn. I would stick with using 90. These turns might have to be provided (children will not know what a 90 turn is)	Children use Beebots in Year 1 but focus mainly on "logical" thinking. The next step in their understanding of algorithms is that they need to be clear/precise/specific with instructions as a machine is not a human and will not understand. Potentially, the children could give instructions to the teacher to do a movement. First, the children would struggle to tell the teacher what to do and the teacher might not understand. The teacher might give the challenge to instruct how to make a jam sandwich. A child may say to the teacher: "Pick up the knife and put it in the butter." The teacher may respond by asking: "Pick up the knife with what?" and would struggle to spread the butter without its lid off. The children would quickly realise that they have to be clearer when programming/instructing otherwise things would go wrong. This understanding of clear instructions is crucial in the next step when they take their algorithms (human language) and input it into a machine (machine language/programming). A machine will not understand an instruction unless it is put into a very clear language specifically designed for that computer. Task (potentially 4-5 sessions): • Children start of the week giving algorithms verbally to teacher/other children to do tasks focusing on clear and precise language. You could maybe take the children into the hall and give them algorithms. E.g. Turn so you are facing the pe cupboard (would a robot know how to do this? No.) • Children can progress onto using machines – Beebots. Consider that the children have already had experience thinking "logically" with these in Year 1. The next step here would be to show a deeper understanding of the clear instructions given. • Finally, children could be pushed into using Turtle Logo to draw shapes. Like before, the algorithm to draw a square might be quite simple to think of. Year 2 have to put this into a machine language (programming) E.g. Algorithm = put the pen down, go forward 5 steps. Programme = pd, fd 5 www.turtleacademy.com/playground www.transum.org/Software/Logo/
Create and debug simple programmes	Depending on the time you teach this, the children will be familiar with algorithms and how they are like instructions. They need to be clear and turned into a programme (machine language). Most programmes do not work as they should first time round; professional programmers have this experience all the time! One of the most rewarding aspects of programming is finding and fixing these mistakes. Mistakes in programmes are	Debugging Scratch Sprite (the character or object) Stage (where the sprite appears) Blocks Coding (the process of building blocks to create a programme)	Children can create and debug simple programmes in Turtle Logo. However, to get an in depth understanding I think that using scratch would be appropriate here. Recap and assess their understanding of what an algorithm/programme is and plug any gaps if needed. You could model how we turn an instruction (algorithm) into a programme. Discuss that things do not always go right when working with programming and that mistakes should be celebrated as they give us an opportunity to make things more efficient/better. Task (potentially 2-3 lessons) • Model use of scratch to accomplish a given goal. Consider that this will be the first time the children have seen Scratch. This goal can be class-led. For example, you could get a sprite to move and play a sound. • Work your way through the blocks, explaining each as you go, to create a programme that the children think will work. Children should be introduced to the term "coding": the process of turning an algorithm into a programme and putting it into a machine. Press the

Featherstone Primary School: Progression and Sequencing within Computing

<table>
<tr>
<td></td>
<td>called ‘bugs’ and finding and fixing them is ‘debugging’.
The process of debugging often involves identifying that there is a fault, working out which bit of the programme (or underlying algorithm) has caused the problem, and then thinking logically about how to fix it. In the classroom, this can provide a great opportunity for collaborative work. It will also help build core values like resilience, independence and persistence. Note: Do not debug a programme for a child straight away. Encourage them to:
1) Think – what is the programme meant to achieve?
2) Which bit/bits are not correct?
3) How can I correct this?
4) Could I look at a programme made by my peer?</td>
<td></td>
<td>green flag (which we call a trigger) and evaluate its success (this would be a perfect opportunity to purposely make a mistake and work collaboratively as a class to “debug” it – share the LO here).
• Differentiation- you may want to give children a select number of blocks that they can use (these could be displayed on the board – Motion, Looks Events)
• You may want to challenge your more able children to add in sounds
• This would be a good opportunity to also teach saving and retrieving (fundamental skills) so that the children can work on this over a number of lessons
• Scratch is available on iPads and on Laptops</td>
</tr>
<tr>
<td colspan="2">Fundamental Use of Technology</td>
<td colspan="2">Technology in the Wider Curriculum
**Note: these are not computing objectives</td>
</tr>
<tr>
<td colspan="2">By the end of Year 2, children will have used technology purposefully to create a digital document. They will have applied skills learnt in Year 1 and Year 2.

• Embedding Year 1 fundamentals
• Drawing shapes
• Shape fill
• Shape outline
• Font (style, colour, size)
• Underlining and italics
• Saving and retrieving a file
• Creating a document</td>
<td colspan="2"><u>Sorting everyday materials (Science) – Pic Collage</u>
You will have to model how to create a sorting chart of different materials in the app Keynote. Start a new project (+button) and create five coloured boxes with text: wood, metal, plastic, rocks and glass. You can get the children to export that image and save it to their camera roll. Alternatively, you can make the sorting chart yourself and “air drop” it to all the children’s iPads. Open PicCollage, change the layout to 4:3 and add your picture into the background. Children can then take photos of different materials and sort them into the categories. If the children know of some things that are made out of that material, they can also do a web search in the app too.

<u>Reading (inference) - Speech bubbles on the Balloon Stickies App</u>
This is a simple to use app that allows children to take photos and add speech bubbles to them. This might work well with a picture book. Children can take a photo of a certain page, add it to the app and then create speech bubbles on the image. They can add a speech bubble and text to infer what characters are thinking at certain points in the story.

  </td>
</tr>
</table>

Featherstone Primary School: Progression and Sequencing within Computing

Year 3 Computing Curriculum Guide

Content Knowledge	Subject Knowledge	Vocabulary	Potential journey/outcome
Use logical reasoning to explain how some simple algorithms works and with support, detect and correct errors in algorithms and programmes.	Children will know what an algorithm/programme is. They may refer to this as an instruction. Deepen the children's understanding by outlining the difference. Algorithm = A set of instructions given to something Programme = A set of instructions given to a machine in "machine" language. Children should understand from previous years that when programming a machine, we have to very clear as they do not understand instructions that are not in their language. Children will also be able to recap why we need to think "logically." E.g. Will a Beebot be able to follow the instruction to look at the door? No. Will a Beebot be able to jump? No. The children will explore this further during the tasks. The main focus of the first objective is to detect and correct errors (this is referred to as debugging).	Debugging Errors Logically Algorithm Programme Coding Stage Background Sprite Scripts	Children to start of by recapping previous knowledge outlined in "subject knowledge" before moving onto debugging. Allow the children to play the game on the following website: bit.ly/pricomp050 Most children will get frustrated with this game as it is not programmed correctly. The programme is wrong in some areas. Can the children find out where? The main error is that the strength button does not work: no matter what is clicked, the strength is always 9. This is an error (a bug in the programme). When the children click "see inside", they will be able to see the code inside (code is just a series of programmes given to a machine). Allow children to explore the code to try and find the mistake. The children may or may not be able to do this. This is fine: the objective says, "with support." Think out loud with the children and discuss through each bit of code. "Year 3, what does this code mean? What does it tell the machine to do?" Go through each one so that the children can understand the programme. Refer to the different parts of the Scratch software: sprite, stage, background, scripts. On each number sprite, the code is incorrect. Allow children to change each number: When this sprite is clicked (number 1), set strength to 1. Repeat this for each number and play the game. Allow children time to reflect on how easy/difficult it was to debug the code (maybe this could be put on post it notes for the learning journal). Spend the next few lessons exploring how to make games with the children on scratch. If you open scratch, click on the tutorials tab- choose the "Make a Chase Game" option. This will take the children through a step-by-step guide on how to make the game. Refer to detecting and correct errors throughout and praise good debugging. Note here that children are also started to write programmes to accomplish given goals (objective 2). You may want to push the children to take their game one step further by adding in effects, colour changes etc. The children explored using sounds in Year 2.
Begin to write programmes to accomplish given goals	Sphero – see "How to Guide" Staff Common > Subject Leaders > ICT Computing > Year 3 Sphero Guide	Block Coding Calibrate Movement Roll Heading	Children will need to get to grips with connecting and calibrating their Sphero before they use them, they will need to do this throughout the lesson independently. First, discuss what the Sphero is. The Sphero is a machine, just like a laptop and scratch. The Sphero follows instructions in a similar way. Why doesn't it follow them exactly the same? Because it might not understand the program (computer language) that Scratch understands and vice versa. Block coding is the process that the children are going to do today. Define this as simply coding, like in Scratch, but in easily understandable blocks. Once the children have calibrated, they can get used to the Sphero by exploring the "draw" and "drive" functions. You may want to spend a whole lesson on this. When comfortable, introduce the LO and discuss what are given goal could be when using a Sphero. Alternatively, give the children an assault course to complete using block coding.

Featherstone Primary School: Progression and Sequencing within Computing

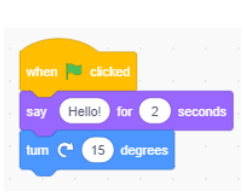
			This may need to be done in the hall and in groups of 6. Encourage the children to plan their code thoroughly before they try. A lot of time will need to be spent on how we can control the Sphero. It does not understand length (as in the children will not have a block to say "roll for one metre"). Children will have to explore the "speed" and how long they roll for before stopping and then changing their "heading" or direction. You may want to show the blocks that the children can use on the hall projector. For example, children, you can only use these certain blocks. Challenge each group to complete the assault course working as a team to write their programme. As the objective states that the children are "beginning" to write their own programmes, you may have to keep each group's assault course the same and slowly reveal a WAGOLL programme that tells the children how to complete it.
Use search technologies effectively and understand how results are shown (selected and ranked)	What is a search engine? Search engines are special websites that make it easy for you to find a website or page in an instant. How do search engines work? To find what you are after, a search engine will scan its index of webpages for content related to your search. A search engine makes this index using a programme called a 'web crawler' or 'web spiders'. This automatically browses the web and stores information about the pages it visits. What is an URL? Every website has its own online address, called a URL. URL stands for Uniform Resource Locator. When you are viewing a page on the World Wide Web, it is the long address that appears in the address bar at the top of your browser. Consider using the Barefoot link below: https://www.barefootcomputing.org/resources/ranking-search-activity You do have to register (for free) but the resource is really good.	URL Search Engine Results Appropriate Algorithm	This objective is based on using the internet to research. Consider watching the video as a whole class (or just for you as subject knowledge) and discuss: https://vimeo.com/44658414 First, the children need to know about staying safe on the internet. The main dangers being: webpages could install rogue software/viruses which will put your computer/laptop at risk and inappropriate content. Discuss the search engines that children use. Most will use google but some may have heard of Bing, Yahoo, Kidrex and Kiddle (Kiddle.co is a safe search engine for kids). Consider watching the YouTube vide on searching safely: https://www.youtube.com/watch?v=kOmSLK42HzI Use this objective to research about something related to the topic: maybe the children could research facts about the rainforest in preparation for a writing task. Things to explore when researching: 1) How does a one-word search compare to a three-word search for effectiveness of results? 2) Are there any key words we could add to our search to make them appropriate? E.g. "For children" or "For kids". 3) How does the order of the words I search effect the results? Which order is best and why? 4) Keywords in your search criteria. You need to think carefully about the keywords you enter in your search. They need to be relevant. For example, to find out how hot it is on the planet Mars, you might search for the words 'Mars', 'surface' and 'temperature'. 5) Using quotation marks in your search will find you that exact phrase. 6) Adding a minus symbol before a word will exclude pages with that word in it. 7) Using Ctrl + f when on a page to find out how many times a word is mentioned. 8) Consider top results against lower down results and why this might be. 9) Consider Google featured snippets 10) Consider search results with AD (advertisements in them). 11) Capital letters when searching. 12) Use of "a" , "an", "the" when searching.

Featherstone Primary School: Progression and Sequencing within Computing

Fundamental Use of Technology		Technology in the Wider Curriculum **Note: these are **not** computing objectives	
• Embedding Year 2 fundamentals • Word Art • Spell check • **B** / U • Copy and paste information/pictures into a document • Re-sizing pictures • Beginning to use Microsoft PowerPoint and adding slides		History Timeline- RWT Timeline app Children can create a Date, Time or Event timeline. You may need to create a profile for this. Children can create labels, short descriptions and then a full description for events. They can also choose their date and add pictures Chatterpix Children can take a picture of anything using the iPad and make it talk using this app. Take a photo, draw a line for the mouth and add eyes and then narrate over the top off the image. The children can use this by taking a photo of Tutankhamun and narrate the mummification process. Alternatively, then can choose a Roman Emperor and explain what Ancient Romans wore etc. Popplet app – Mind mapping A great tool for mind mapping anything. Children can add pictures and text to an electronic mind map to collate ideas. Kahoot Quiz Create a quiz for anything. Create a quiz online, display the code and then let the children join on the iPads. Children work through a series of multiple-choice questions where they will be scored on correct answers and their speed. This is great for retrieval questions in reading or as a non-core plenary to assess understanding.	 

Featherstone Primary School: Progression and Sequencing within Computing

Year 4 Computing Curriculum Guide

Content Knowledge	Subject knowledge	Vocabulary	Potential journey/outcome
Use logical reasoning to explain how some simple algorithms works and with growing independence, detect and correct errors in algorithms and programmes Consolidate and embed how to write programmes to accomplish given goals	Algorithm = A set of instructions given to something Programme = A set of instructions given to a machine in “machine” language. Coding = The process of writing the machine language See vocabulary for Python codes. Children need to ensure that they have the correct “syntax" for Python. Syntax refers to how a code is built (in terms of capital letters and punctuation). For example, the code hero.moveUp() will work, but the code hero.moveup() will not work. Codes for Python (Level 1-10) hero.moveUp() hero.moveDown() hero.moveLeft() hero.moveRight() hero.say(“Hello”) hero.attack(“Bob”) hero.findNearestEnemy() while True: Inside the brackets is a “variable”. E.g. How many times does that code need to be done?	Python Syntax Coding Debugging	<u>Python coding</u> Recap previous year group objectives by discussing the difference between an algorithm/programme and what coding is. Algorithm = A set of instructions given to something Programme = A set of instructions given to a machine in “machine” language. Coding = The process of writing the machine language Start of by discussing what the children have programmed so far: Scratch, BeeBot and Sphero. Ask what type of programming you would give to each. Would a BeeBot understand this coding block? (see fig.1) Would a Sphero understand this? (see fig.2) You may want to have an example ready on each table (Turtle Logo, Sphero, Scratch and BeeBot) for the children to use and explore.  Fig.1  Fig.2 Essentially, the children have to understand that different machines use different programming languages. It is a little bit like speaking French to an English person who does not know French. Children will be experimenting in these sessions with “Python”- experience with different coding languages is essential. Start by playing a game of “Simon Says” to get the children thinking about Python language. Explain to the children that you are going to be programing them with the codes. Like in Simon Says, the children should only do the action if it starts with “children”. Continue by giving the children instructions like: Children.Turn Right (they will turn right) and Turn right (children should not move). Tell the children that this is just like coding Python. You have to start by coding with the thing you are coding. Children to access www.codecombat.com and click “Play” Levels 1-10 use the codes above (see subject knowledge).Encourage children to debug their mistakes as they go. Plenary: How is Scratch different to Python/Turtle Logo/Sphero Which is easier to use? Finish by playing another game of “Simon Says” using coding language.

Featherstone Primary School: Progression and Sequencing within Computing

<table>
<tr>
<td>Use sequence, selection and repetitions in programmes and work with variables and various forms of input/output</td>
<td>See the maze game that I have created in the Subject Leader >>> Computing folder.</td>
<td>Block Coding
Debugging
Sequence
Selection
Repetition

Scratch Block Codes used:
• Motion
• Looks
• Sound
• Event
• Control
• Sensing
• Variables</td>
<td><u>Scratch Gaming</u>
Children will continue their “gaming” by designing their own game on Scratch using “block coding”. Alternatively, the children could create their own quiz. It is up to the teacher what fits best and what will interest the children more. Note: Both units of work can be found in the Subject Leader > Computing folder. You can find an example of a “maze game” on the following link: https://scratch.mit.edu/projects/57805794/ Play this on the IWB with the children and discuss what codes need to be in place. Examples from the children could include:
1) Code to move the sprite using the arrow keys
2) Code to “bounce” the sprite back if it hits the colour red
3) Code to make the sprite say “well done” when you hit the colour green
4) Maybe there could be code to take you to a different level after

The children will then need to draw their game (with 2 possible levels). After drawing, the children can then annotate each part with the input and output. E.g. “If I hit the red block (input), then the sprite will bounce back (output)” or “If the sprite reaches the green square (input), the sprite will say well done and the next level will appear.” Decomposing it in this way will allow the children to explore “computational thinking.” In essence, if the children know what needs to happen step-by-step with every part of the game, their coding will be easier.

To make the game, the children will have to follow these steps:
1) Resize their sprite (use the costumes tab – make sure the sprite stays in the middle of the page here otherwise it will not work).
2) Design a backdrop (using the rectangle shape tool to draw the maze obstacles – make sure the colour of the maze obstacles stays the same).
3) Make a more complex new backdrop for level 2 (if appropriate). Children could also add another sprite here as an “enemy” for the sprite – if the sprite touches the enemy, it will have to go back to the start.
4) Add effects to the game (sound or speech).
5) Add a variable – a piece of code that changes in value (this could be used to keep score of how many times they hit the maze or how many levels they get to).
In each step, the children will be programing using sequence, selection and repetition. For example, to successfully program a game, they will need the “Forever” or “When something is pressed, this will happen” code.

I have created a WAGOLL maze game that works perfectly and has comments in it to help with what each code does.</td>
</tr>
<tr>
<td colspan="2">Fundamental Use of Technology</td>
<td colspan="2">Technology in the Wider Curriculum
**Note: these are not computing objectives</td>
</tr>
<tr>
<td colspan="2">• Embedding Y3 skills
• Copy and paste using keyboard shortcuts (Ctrl + C and Ctrl + V)
• Creating a folder
• “Dragging and dropping” a file into a folder
• Using Microsoft PowerPoint to add animations/transitions</td>
<td colspan="2"><u>Geography (Rivers) – WWF Free Rivers app</u>
Children can use the app to investigate the importance of rivers using AR (augmented reality). You can interact with the AR river which will teach the children all about rivers around the world, including dams. It can also take the children on a tour of the river’s journey that you can zoom into.

<u>Geography (UK Counties) – ThingLink app</u>
Children can use a map of the UK and link things to each county. The children can tap anywhere on the photo and embed something onto it. This can be narration of what you can find in this area, text or even photos.
  </td>
</tr>
</table>

Featherstone Primary School: Progression and Sequencing within Computing

	Popplet app – Mind mapping A great tool for mind mapping anything. Children can add pictures and text to an electronic mind map to collate ideas. Kahoot Quiz Create a quiz for anything. Create a quiz online, display the code and then let the children join on the iPads. Children work through a series of multiple-choice questions where they will be scored on correct answers and their speed. This is great for retrieval questions in reading or as a non-core plenary to assess understanding.

Featherstone Primary School: Progression and Sequencing within Computing

Year 5 Computing Curriculum Guide

Content Knowledge	Subject knowledge	HTML Code	Potential journey/outcome
Design, write and debug programmes that accomplish given goals, incl. controlling or simulating physical systems; begin to solve more complex problems by breaking them down into smaller parts, with support or advice	Consider using this link to structure an introduction/input to what the internet is. The children study this in KS1 but the following website gives an insight into the more complex systems used. https://www.bbc.co.uk/bitesize/topics/z7wtb9q/articles/z3tbgk7 HTML All web pages on the internet are created using a language called Hypertext Markup Language (HTML). HTML describes: • what information appears on a webpage • how it appears on the page (formatting) • any links to other pages or sites HTML can be written in specialist software, or in a simple text editor like Notepad. As long as the document is saved with the file extension '.html' it can be opened and viewed as a webpage from a browser. This example HTML code displays a message on a webpage: <html> <body> <h1>Hello world</h1> This is my first webpage </body> </html> The code uses tags to describe the appearance of the information: <html> states that the document is a HTML document <body> states that the information appears in the body of the page <h1> states that the following text appears as a prominent heading states that this is the beginning of a new paragraph	<!DOCTYPE html> This is heading 1 This is heading 2 This is heading 3 <title>My first web page</title> This is a paragraph. This is another paragraph. This is a link - First thing - Second thing - Third thing (Numbered list) - First item - Second item - Third item (Bullet point list) There are many more HTML codes to use. Consider looking at https://www.w3schools.com/html/default.asp for a guide on how to use HTML codes. Alternatively, www.htmldog.com provides a similar service.	Design Inform the class that they will be designing and creating their own websites (linked to "Help for Heroes" or WW1/WW2). Display a generic website on the IWB, maybe the school website, and press Ctrl + U – this will bring up the source code for the site. You may want children to do this on their own laptop and browse different websites. Explain that behind every website, there is code. However, there are many ways to code a website. With the children, consider what the first website might have looked like. Visit: http://info.cern.ch and click on either the first two links. This is a reconstruction of what the first website was. Use Ctrl + U to compare the code to a modern website. Encourage the children to think about how the Web might evolve in the future. Create a design criteria based on this for a successful website. You may want to get the children to draw their website. After drawing, they can annotate their website. E.g. This part needs to be underlined. This part needs to be in bold. This part needs to be red. A picture will go here. This will allow the children to "decompose" their website/coding into smaller parts and then pick and choose which HTML coding they will need to accomplish the design. Write and debug There are many platforms that you can use to write your own code for a website. Other than using a basic notepad, you can use: • www.codecademy.com • www.glitch.com (I like this one as you can see your website live as you code). • www.sites.google.com (easier versions of above) • https://jsfiddle.net/ Alternatively, these sites can be a good modelling tool. If you use notepad, you will be able to have a lot more control of what the children do, rather than letting them discover at their own pace. Ensure that the website designing is broken down into "smaller parts" as mention in the objective, to ensure that the children do not get overwhelmed with code. HTML Code can be given to the children (either on IWB or "cheat sheets") as the objective states "with support or advice". Consider peer-assessment on websites before publishing them (link to showcase here).

Featherstone Primary School: Progression and Sequencing within Computing

Use and combine given software, on a range of devices, to accomplish given goals (collect, analyse, design, create, present and analyse)	Animation creates the impression of movement through an optical illusion referred to as the "Persistence of Vision." The eye retains an image for a split second after it has actually been shown. Animation works by presenting slightly different images in quick succession, with the persistence of vision filling in the gap between each image and allowing for the illusion of motion. In the 19th century, this phenomenon was made use of in many children's toys that some students may still be familiar with. In the 19th century, this phenomenon was made use of in many children's toys that some students may still be familiar with: • Thaumatropes • Mutoscopes • Flick books • Zoetropes	Stop motion Frames Per Second Onion Layering Animation Set Props Lighting	Introduce the aims of the next few computing sessions. The children will be using a range of **given** software to create a stop motion animation. They will be using a maximum of two apps to use. Stop Motion App, to take photos, and movie maker, to add titles/music etc. Show children examples of stop motion. Many can be found on YouTube, but the links below are quite good. • https://www.youtube.com/watch?time_continue=56&v=PhKqjevqTtE&feature=emb_logo • https://www.youtube.com/watch?time_continue=43&v=3p-xDZ0Ra-s&feature=emb_logo • https://www.youtube.com/watch?v=K87ANkZBIY8 Discuss with the children which one they prefer. They may notice that the Simpsons video does not seem as good. See if the children understand the reason for this. Go through the history of stop motion animation, making references to the children's toys that are mentioned in the subject knowledge column. The stop motion animation can be made with any purpose in mind: re-telling your current reading book, re-enacting a scene studied in history or even to tell a story about respect in HMHB sessions. Consider watching the following videos which explain stop motion further: http://www.bbc.co.uk/learningzone/clips/stop-motion-explained-by-an-aardman-animator/10151.html and http://www.kameleon.co.uk/news/sonys-guide-to-stop-motion-animation-was-launched-across-europe.html. I suggest the journey of the topic follows this: 1) Introduction to stop motion 2) Creating my own flipbook (to understand the idea of repetition with minimal movement) 3) Experimenting with "Stick Nodes" app and understanding what "Onion layering" is – the idea of creating layers after layers. 4) Creating their own stop motion drawing with simple pen and paper https://www.youtube.com/watch?v=Xo2ioUYugMA . 5) Storyboard to plan animation 6) Creating their own stop motion with clay using the stop motion app on iPads (discussing things like keeping the iPad steady, sound quality, set, lighting, props etc). 7) Combining software – saving their stop motion and then adding it into movie maker to add a title slide/music etc (this is where the actual computational thinking objective comes into this). 8) Peer-assessing Top tip – The children will not take enough photos of each step. This will mean that the animation will seem very jerky and will play too quickly. Encourage the children to take multiple photos of each scene, especially if writing is involved. Also, consider changing the FPS (Frames Per Second) which will change the number of photos that are played per second.

Featherstone Primary School: Progression and Sequencing within Computing

Fundamental Use of Technology	Technology in the Wider Curriculum **Note: these are **not** computing objectives
• Embedding Y4 skills • Alt + Shift to alternate between windows • Drawing a table and inserting columns/rows • Bullet-points and text alignment • Formatting pictures • Applying all skills in a Microsoft document	Timelines - Adobe Spark Video Create a timeline to show the events that led to the cause of WW1. Children can use this app to make a short video, using photos/videos/caption and even their own narration. The app is really easy to use and can be done to assess the children's knowledge of chronology and key events. Geography/Maps - Keynote Alternatively, children can use a map of Europe and Keynote to show the countries that were involved in the war. They can create an animation to show how Germany invaded other European countries and how Britain and her allies responded. Google Expeditions Explore the moon using Google Expeditions virtual reality. Google Expeditions will immerse the children into any environment and can be used for a number of topics. AR Moon app Take the children outside to use this augmented reality moon app. The app goes through launching the Apollo space shuttle into space and then complete different stages to explore the different parts of the moon. It will set challenges for the children to drive the lunar module and collect pictures. Musical score- Garage Band and iMovie/Bossjock Garage band allows children to use many instruments to create their own "musical score". For example, the children can use the string instrument option and click on "auto play" which will produce an automatic melody. The children can add layers onto this using other instruments and create a soundtrack. The soundtrack can they be saved or exported into iMovie or Bossjock. The children can then read a story out loud and use the soundtrack to accompany it and create suspense/atmosphere. This might work well if the children have created their own stories or if you want to work on reading out loud – maybe even the Highwayman poem.

Featherstone Primary School: Progression and Sequencing within Computing

Year 6 Computing Curriculum Guide

<table>
<tr><th>Objective</th><th>Subject knowledge</th><th>Vocabulary</th><th>Potential journey/outcome</th></tr>
<tr>
<td>Design, write and debug programmes that accomplish intended goals chosen by them, incl. controlling or simulating physical systems; solve more complex problems by breaking them down into smaller parts</td>
<td>This objective is linked to the DT objective of using computing to control and stimulate physical systems/products.</td>
<td>See Crumble How to Guide</td>
<td>Robot Wars
Children to follow the Crumble How To Guide during this journey.
Step 1 – Introduce “Robot Wars” and aim of this unit – to build a robot that can be controlled with computer coding to battle against other robots.
Step 2- Design phase – children to design a robot following DT objectives (research, design, prototype, exploded diagram)
Step 3- Build robot out of chosen material
Step 4- Computer coding using “Crumble”. Children to block code their Crumble for movement (turn left, right, move forward and reverse). Crumble is a scratch-type coding programme.
Step 5- Children to compete with their robots over a certain amount of events, such as:
• Timed sprint, quickest bot from A to B wins
• Timed obstacle course
• Best decorated robot
• Knock out tournament where robots will progress through either destroying the opposition or (in judges’ opinion) inflicting more damage (hits) than its opponent.
Step 6- Evaluate</td>
</tr>
<tr>
<td>Select, use and combine given software, on a range of devices, to accomplish given goals</td>
<td>The progression of this objective from year 5 is the use of the word “select”. In Year 5, the children are given the software to use (Stop Motion App and Movie Maker). For Year 6, the next step for the children is to choose their own software. You may need to spend some time using the potential software needed. Various apps that can be used here are:
• DropBox
• Keynote
• iMovie
• Filmmaker Pro
• Clips</td>
<td>Composition
Focus
Effects
Editing
Landscape
Portrait
Angles
Digital devices
Cropping
Software</td>
<td>Introduce the objective: in this Computing journey we are going to be creating our own “Year 6 Tribute Movie” to celebrate the end of Year 6 and our transition into secondary school.
Explain the objective by discussing that the children are going to be selecting their own software to use. Discuss what the goal is – to make a video to show at our production about the amazing time we have had at Featherstone/to say thank you to our teachers etc. Look at the iPad apps with the children and discuss which apps can be used to do all the things we need:
• Taking photos
• Recording videos
• Editing both of the above
• Adding title pages
• Adding writing
• Using music to add atmosphere
• Saving the video online so that we can access it
Clearly, the children will not be able to use one app to do all these things.

Let the children explore the apps by giving them a task: record and edit a short video (no longer than 30 seconds) in your group about your favourite Year 6 memory. Remind the children that this lesson is all about exploring the app and experimenting with its features. Praise the children that are trying different apps to see which one is the best. You may want to design a document where the children can evaluate the apps. E.g.

<table>
<tr><th>Apps</th><th>Add music</th><th>Add effects</th><th>Easy to use</th><th>Video quality</th></tr>
<tr><td>iMovie</td><td>Y</td><td>Y</td><td>N</td><td>Y</td></tr>
<tr><td>Clips</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td></tr>
<tr><td>Filmmaker pro</td><td>N</td><td>N</td><td>N</td><td>Y</td></tr>
</table>
</td>
</tr>
</table>

Featherstone Primary School: Progression and Sequencing within Computing

			Discuss with the children the idea photography and filmography. Look at bad and good examples of both. Go through examples of composition, background, foreground, focus, effects (see vocabulary) etc and give the children a small task – take a photo of this object. Allow the children to choose an object and explore taking photos of it. You may want to make this into a competition. Let the children film and edit their tribute video for the next few lessons, focusing on quality of video and photography. Allow the children to be involved in discussing the different parts of the video. E.g. Interviews with previous teachers, photos of the whole class, best Featherstone bits etc. Show the children how to upload their finished videos to Dropbox and evaluate the effectiveness of each with the class.
Fundamental Use of Technology		Technology in the Wider Curriculum **Note: these are **not** computing objectives	
• Embedding Y5 skills • Manipulate windows for split screen (when combining software) • Watermark • Page insert/orientation • Embed table knowledge including merging cells • Applying all skills learnt in a Microsoft PowerPoint		<u>Mentimeter app – vocabulary</u> Children join with your code on the iPad. They can scan a text for any vocabulary they do not understand and input it onto their iPad. The words will create a word cloud on the IWB. The more inputs, the bigger the word. Screenshot the mind map and go through the words on Active Inspire. This can be saved and printed for a working wall. <u>Apple clips</u> Children to create short videos on any subject topic. They can narrate, add text, add effects and photos to the video. <u>Popplet app – Mind mapping</u> A great tool for mind mapping anything. Children can add pictures and text to an electronic mind map to collate ideas. <u>Kahoot Quiz</u> Create a quiz for anything. Create a quiz online, display the code and then let the children join on the iPads. Children work through a series of multiple-choice questions where they will be scored on correct answers and their speed. This is great for retrieval questions in reading or as a non-core plenary to assess understanding.